BITS F464 Machine Learning

Assignment 1

Date : 15th March, 2020

Team Members

Durba Satpathi	2019A7PS0972H
R Adarsh	2019A7PS0230H
Navdeep Singh Narsinghia	2017B5A71675H

Problem 1B Naive Bayes Classifier

2.1. Model Description and implementation

Naive Bayes is a statistical classification technique based on Bayes Theorem. We build a probabilistic machine learning model that is used for classification tasks. Our classification model produces the probability of mails to be spam or not spam by the condition of given words.

Dataset Given:
The dataset used for this assignment contains the content of the mails followed by the target variable indicating if the mail is spam or not (1/0).

Preprocessing:
In general, it is a good idea to preprocess the given raw data and remove irrelevant parts of text that do not contribute to the sentiment of the text. The following are the preprocessing steps used:

1) Removal of punctuation marks , special characters and numbers from the text.
2) Converting the entire text to lower case.
3) Removal of trivial one letter and two letter words.

Cross Validation:
The dataset is split into k(=7) groups (7-fold cross validation). For each unique group, consider it as the test set and remaining groups as a train set. Fit the model on the train set and evaluate on the test set and obtain the evaluation metrics. Finally we summarize the overall evaluation score using the sample of model evaluation metrics.

Dataset Preparation:
We have made the dataset using a python dictionary which contains information about occurrence of each word in each category along with its frequency.
Format: {('word', 'category(0/1)'): freq }

Procedure:

1) If the word is present in our prepared vocabulary then we compute the probability as follows:

$$P(w_i|class) = \frac{freq(w_i,class)+\alpha}{N_{class}+\alpha V}$$

where

- $P(w_i|class)$: *probability of occurrence of a word* w_i *given a class* :
- $freq(w_i|class)$: *frequency of occurrence of a word* w_i *in a given class*
- α : *smoothing parameter*
- V : *number of unique words in the vocabulary*

- N_{class} : *frequency of all words in a class*

If the word is not present in the vocabulary, we calculate the probability by assigning the frequency of the word as 0.

2) Given the testing data we compare the probability of data being in category 1 and the probability of data being in category 0.

$$P(w_1, w_2, ..., w_n|class) = \frac{P(w1,w2,.....,wn|class)P(class)}{P(w1,w2\ldots,wn)}$$

$$= \frac{P(w1|class)P(w2|class).....P(wn|class)P(class)}{P(w1)P(w2)\ldots P(wn)}$$

(assuming the occurrences of words are conditionally independent of each other)

where

- *P(class |w1,w2,....wn) i.e* posterior probability
- *P(w1,w2,....wn|class) i.e* likelihood
- P(wi|class)=conditional probability of occurrence of the word wi given class
- P(class)=probability of occurrence of the class (Class Prior Probability)
- P(w)=probability of occurrence of the word w (Predictor Prior Probability)

Note: It is sufficient to compare the numerator as the denominator would remain the same.

3) Naive Bayes classifier combines the model with the decision rule. One common rule is to pick the hypothesis that's most probable.

$$class = argmax_{class} P(class) \prod_{i=1}^{n} P(word_i|class)$$

2.2. Accuracy of model over each fold and the overall average accuracy

```
Accuracy over fold  1 :    80.98591549295774 %
Accuracy over fold  2 :    86.01398601398601 %
Accuracy over fold  3 :    82.51748251748252 %
Accuracy over fold  4 :    81.81818181818181 %
Accuracy over fold  5 :    80.41958041958041 %
Accuracy over fold  6 :    82.51748251748252 %
Accuracy over fold  7 :    78.32167832167832 %
Overall average accuracy:  81.7991867287642 %
```

We have achieved overall accuracy of around 82%.

2.3. Major limitations of the Naive Bayes classifier:

1) The major limitation of Naive Bayes is the assumption of independence of attributes. In real life datasets, it is almost impossible to achieve a set of predictors which are completely independent. This is the problem of the "Naive Assumption".

2) If the test set has a category that is not present in the train set then the model would assign 'zero probability' and hence cannot make a conclusive prediction. This happens when we are drawing samples from a population and the drawn vectors are not fully representative of the population. This is known as the problem of "Zero Frequency".However, in our implementation, Laplace smoothing solves this by giving the last word a small non-zero probability for both classes, so that the posterior probabilities don't suddenly drop to zero.

3) It treats all attributes equally and doesn't account for the semantic relationship between the words. But this may not always be true as there may be some attributes words that have greater contribution towards the final decision than others.